



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/667,542	09/22/2003	Robert Anthony DeLine	MS304414.1 /MSFTP464US	9926
27195	7590	01/10/2008	EXAMINER	
AMIN, TUROCY & CALVIN, LLP 24TH FLOOR, NATIONAL CITY CENTER 1900 EAST NINTH STREET CLEVELAND, OH 44114			YIGDALL, MICHAEL J	
			ART UNIT	PAPER NUMBER
			2192	
			NOTIFICATION DATE	DELIVERY MODE
			01/10/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docket1@thepatentattorneys.com
hholmes@thepatentattorneys.com
osteuball@thepatentattorneys.com

Office Action Summary	Application No.	Applicant(s)	
	10/667,542	DELINE ET AL.	
	Examiner	Art Unit	
	Michael J. Yigdall	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 15 November 2007.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-7 and 9-25 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-7 and 9-25 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413) Paper No(s)/Mail Date. _____
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	5) <input type="checkbox"/> Notice of Informal Patent Application
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____	6) <input type="checkbox"/> Other: _____

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on November 15, 2007 has been entered. Claims 1-7 and 9-25 are pending.

Response to Amendment

2. The rejection of claims 1-16, 22 and 23 under 35 U.S.C. 101 has been withdrawn in view of Applicant's amendment.

Response to Arguments

3. Applicant's arguments (see Applicant's remarks, pages 9-16) with respect to the rejections of the claims under 35 U.S.C. 103(a) have been fully considered and are persuasive. Accordingly, the rejections have been withdrawn. However, upon further consideration, new grounds of rejection are made as set forth below.

Claim Objections

4. Claims 22 and 23 are objected to because each claim recites the term "trasmitting" rather than "transmitting." Appropriate correction is required.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

6. Claims 1-7 and 11-25 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 7,036,111 to Dollin et al. (now made of record; “Dollin”).

With respect to claim 1 (currently amended), Dollin teaches an executable code check computing system (see, for example, FIG. 1 and the abstract) comprising:

an input component operating on computer hardware that receives an executable object file having an embedded specification, the specification specified at a source code level (see, for example, column 3, lines 19-24, which shows receiving an executable object file in the form of a compiled program, and see, for example, column 5, lines 20-25, which shows translating the instructions of the program into type signatures, and column 5, lines 26-34, which shows that the type signatures represent a specification).

The type signatures are considered an “embedded specification” at least because they are derived from information embedded in the program. The embedded specification is “specified at a source code level” in the sense that the instructions from which the type signatures are derived are first specified in the source code of the program.

Dollin further teaches:

a checker operating on computer hardware that employs the specification to facilitate static checking of the executable object file, the checker providing information if a fault condition is determined, the fault condition is based on one or more of a violation of rules for using an interface, system resource management rules, rules for proper ordering of method calls, or string parameter format rules (see, for example, column 3, lines 25-31, which shows a code verifier that employs the type signatures to facilitate static checking of the program, and column 16, lines 45-54, which shows that the code verifier provides information if a fault condition is detected, and see, for example, column 3, lines 5-13, which shows that the fault condition is based on a violation of rules such as those for using an interface, and column 16, line 66 to column 17, line 6, which shows rules such as those for proper ordering of method calls).

With respect to claim 2 (original), the rejection of claim 1 is incorporated, and Dollin further teaches the checker further removing the embedded specification from the object file (the specification is “removed” from the program in the sense that the type signatures are extracted from the program and are not persisted in the program after code verification).

With respect to claim 3 (original), the rejection of claim 1 is incorporated, and Dollin further teaches the specification comprising information associated with a method that performs at least one of allocation and release of a resource (see, for example, column 16, line 66 to column 17, line 6, which shows that the type signatures comprise information associated with a method, and column 5, lines 39-53, which shows that the method includes an instruction that allocates a resource).

With respect to claim 4 (original), the rejection of claim 1 is incorporated, and Dollin further teaches the specification comprising information associated with an order in which methods of an object can be called (see, for example, column 16, line 66 to column 17, line 6, which shows that the type signatures comprise information associated with the order of method calls in the control flow of the program).

With respect to claim 5 (original), the rejection of claim 4 is incorporated, and Dollin further teaches that method order is constrained by specifying a finite state machine in which the states have symbolic names and transitions between states are labeled with method names (see, for example, column 17, lines 7-26, which shows a finite state machine that constrains method order in terms of transitions from input type constraints to output type results, and column 5, line 66 to column 6, line 14, which shows that the states have symbolic names).

With respect to claim 6 (original), the rejection of claim 1 is incorporated, and Dollin further teaches the specification comprising a state-machine protocol wherein a method specifies a pre-state and a post-state (see, for example, column 17, lines 7-26, which shows that the specification comprises a state-machine protocol wherein a method specifies a pre-state in the form of input type constraints and a post-state in the form of output type results).

With respect to claim 7 (original), the rejection of claim 1 is incorporated, and Dollin further teaches the specification comprising information associated with a transition of a finite state machine (see, for example, column 17, lines 7-26, which shows that the specification

comprises information associated with a transition of a finite state machine such as the transition from the input type constraints of a method to the output type results of the method).

With respect to claim 11 (original), the rejection of claim 1 is incorporated, and Dollin further teaches the specification comprising information associated with a state-machine protocol (see the rejection of claim 6 above).

With respect to claim 12 (original), the rejection of claim 1 is incorporated, and Dollin further teaches the specification comprising an attribute associated with at least one of a field and a parameter providing information associated with whether or not the at least one of a field and a parameter can be aliased (see, for example, column 9, lines 29-37, which shows that the specification comprises attributes associated with fields or parameters that provide information associated with whether or not the fields or parameters can be aliased).

With respect to claim 13 (original), the rejection of claim 1 is incorporated, and Dollin further teaches that the specification facilitates modeling of a heap modeling (see, for example, column 5, line 66 to column 6, line 14, which shows that the specification facilitates modeling of a heap modeling such as with addresses of variables and descriptions of values pushed to the stack).

With respect to claim 14 (original), the rejection of claim 13 is incorporated, and Dollin further teaches the checker employing an algorithm that performs a data flow analysis over the heap model comprising a typing environment and a set of capabilities (see, for example, column 7, line 64 to column 8, line 8, which shows that the code verifier employs such an algorithm).

With respect to claim 15 (currently amended), Dollin teaches an executable code check computing system (see, for example, FIG. 1 and the abstract) comprising:

an input component operating on computer hardware that receives an object file (see, for example, column 3, lines 19-24, which shows receiving an object file in the form of a compiled program);

a checker operating on computer hardware that employs a specification associated with the object file to facilitate static checking of the object file, the checker providing information if a fault condition is determined, the specification specified at a source code level and stored in a specification repository (see, for example, column 3, lines 25-31, which shows a code verifier that employs type signatures to facilitate static checking of the program, and column 16, lines 45-54, which shows that the code verifier provides information if a fault condition is detected, and see, for example, column 5, lines 20-25, which shows translating the instructions of the program into the type signatures, and column 5, lines 26-34, which shows that the type signatures represent a specification).

The type signatures are “stored in a specification repository” in the sense that they are stored in memory. The specification is “specified at a source code level” in the sense that the instructions from which the type signatures are derived are first specified in the source code of the program.

With respect to claim 16 (original), the rejection of claim 15 is incorporated, and Dollin further teaches the specification repository (see, for example, FIG. 1).

With respect to claim 17 (original), the claim is directed to a method of facilitating static checking of executable code, the elements of which are analogous to elements recited in claim 1 (see the rejection of claim 1 above).

With respect to claim 18 (original), the rejection of claim 17 is incorporated, and Dollin further teaches removing the embedded specification from the executable code (see the rejection of claim 2 above).

With respect to claim 19 (original), the claim is directed to a computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 17 (see the rejection of claim 17 above).

With respect to claim 20 (currently amended), the claim is directed to a method of facilitating static checking of executable code, the elements of which are analogous to elements recited in claim 15 (see the rejection of claim 15 above).

With respect to claim 21 (original), the claim is directed to a computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 20 (see the rejection of claim 20 above).

With respect to claim 22 (currently amended), Dollin teaches a computer-readable medium comprising computer executable instructions for transmitting a data packet transmitted between two or more computer components that facilitates static checking of executable code (see, for example, FIG. 1 and the abstract), the data packet comprising:

executable code having an embedded specification, the embedded specification providing information to be employed to statically check the executable code (see, for example, column 3, lines 19-24, which shows executable code in the form of a compiled program, and see, for example, column 5, lines 20-25, which shows translating the instructions of the program into type signatures, and column 5, lines 26-34, which shows that the type signatures represent a specification that provides information for statically checking the program).

The type signatures are considered an “embedded specification” at least because they are derived from information embedded in the program.

With respect to claim 23 (currently amended), Dollin teaches a computer-readable medium comprising computer executable instructions for transmitting a data packet transmitted between two or more computer components that facilitates static checking of executable code (see, for example, FIG. 1 and the abstract), the data packet comprising:

a source-level specified specification that provides information to be employed to statically check the executable code (see, for example, column 3, lines 19-24, which shows executable code in the form of a compiled program, and see, for example, column 5, lines 20-25, which shows translating the instructions of the program into type signatures, and column 5, lines 26-34, which shows that the type signatures represent a specification that provides information for statically checking the program).

The specification is “source-level specified” in the sense that the instructions from which the type signatures are derived are first specified in the source code of the program.

With respect to claim 24 (original), the claim is directed to a computer readable medium storing computer executable components of an executable code check system, the elements of which are analogous to elements recited in claim 1 (see the rejection of claim 1 above).

With respect to claim 25 (original), the claim is directed to an executable code check system, the elements of which are analogous to elements recited in claim 1 (see the rejection of claim 1 above).

7. Claims 1-7 and 9-25 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 7,150,008 to Cwalina et al. (now made of record; "Cwalina").

The applied reference has a common assignee with the instant application. Based upon the earlier effective U.S. filing date of the reference, it constitutes prior art under 35 U.S.C. 102(e). This rejection under 35 U.S.C. 102(e) might be overcome either by a showing under 37 CFR 1.132 that any invention disclosed but not claimed in the reference was derived from the inventor of this application and is thus not the invention "by another," or by an appropriate showing under 37 CFR 1.131.

With respect to claim 1 (currently amended), Cwalina teaches an executable code check computing system (see, for example, FIG. 2 and the abstract) comprising:

an input component operating on computer hardware that receives an executable object file having an embedded specification, the specification specified at a source code level (see, for example, steps 301 and 302 in FIG. 3, which shows receiving a rule assembly and a target assembly, and column 8, lines 20-21, which shows that the assemblies are executable object

files, and see, for example, column 8, lines 28-36, which shows that the assemblies have an embedded specification in the form of rules, and column 10, lines 34-40, which shows that the rules are specified at a source code level); and

a checker operating on computer hardware that employs the specification to facilitate static checking of the executable object file, the checker providing information if a fault condition is determined, the fault condition is based on one or more of a violation of rules for using an interface, system resource management rules, rules for proper ordering of method calls, or string parameter format rules (see, for example, column 16, lines 46-62, which shows employing the rules to facilitate static checking of the assemblies and providing information if a rule is violated, and column 8, lines 37-65, which shows that the rules include rules for using an interface, system resource management rules, rules for proper ordering of method calls and string parameter format rules).

With respect to claim 2 (original), the rejection of claim 1 is incorporated, and Cwalina further teaches the checker further removing the embedded specification from the object file (the specification is “removed” from the assemblies in the sense that the rules are extracted from the rule assembly and are not persisted in the target assembly).

With respect to claim 3 (original), the rejection of claim 1 is incorporated, and Cwalina further teaches the specification comprising information associated with a method that performs at least one of allocation and release of a resource (see, for example, column 8, line 37 to column 9, line 30).

With respect to claim 4 (original), the rejection of claim 1 is incorporated, and Cwalina further teaches the specification comprising information associated with an order in which methods of an object can be called (see, for example, column 8, line 37 to column 9, line 30).

With respect to claim 5 (original), the rejection of claim 4 is incorporated, and Cwalina further teaches that method order is constrained by specifying a finite state machine in which the states have symbolic names and transitions between states are labeled with method names (see, for example, column 8, line 37 to column 9, line 30).

With respect to claim 6 (original), the rejection of claim 1 is incorporated, and Cwalina further teaches the specification comprising a state-machine protocol wherein a method specifies a pre-state and a post-state (see, for example, column 8, line 37 to column 9, line 30).

With respect to claim 7 (original), the rejection of claim 1 is incorporated, and Cwalina further teaches the specification comprising information associated with a transition of a finite state machine (see, for example, column 8, line 37 to column 9, line 30).

With respect to claim 9 (currently amended), the rejection of claim 1 is incorporated, and Cwalina further teaches the object file being based, at least in part, upon a language that compiles to Common Language Runtime (see, for example, column 8, lines 6-10, which shows that the assemblies are based on a language that compiles to a Common Language Runtime such as with a Microsoft .NET compiler).

With respect to claim 10 (previously presented), the rejection of claim 1 is incorporated, and Cwalina further teaches the object file being based, at least in part, upon at least one of C#, VISUAL BASIC.NET and Managed C++ (see, for example, column 8, lines 6-10, which shows that the assemblies are based on a language such as C#, Visual Basic.NET and managed C++).

With respect to claim 11 (original), the rejection of claim 1 is incorporated, and Cwalina further teaches the specification comprising information associated with a state-machine protocol (see, for example, column 8, line 37 to column 9, line 30).

With respect to claim 12 (original), the rejection of claim 1 is incorporated, and Cwalina further teaches the specification comprising an attribute associated with at least one of a field and a parameter providing information associated with whether or not the at least one of a field and a parameter can be aliased (see, for example; column 8, line 37 to column 9, line 30).

With respect to claim 13 (original), the rejection of claim 1 is incorporated, and Cwalina further teaches that the specification facilitates modeling of a heap modeling (see, for example, column 8, line 37 to column 9, line 30).

With respect to claim 14 (original), the rejection of claim 13 is incorporated, and Cwalina further teaches the checker employing an algorithm that performs a data flow analysis over the heap model comprising a typing environment and a set of capabilities (see, for example, column 8, line 37 to column 9, line 30).

With respect to claim 15 (currently amended), Cwalina teaches an executable code check computing system (see, for example, FIG. 2 and the abstract) comprising:

an input component operating on computer hardware that receives an object file (see, for example, steps 301 and 302 in FIG. 3, which shows receiving a rule assembly and a target assembly, and column 8, lines 20-21, which shows that the assemblies are executable object files);

a checker operating on computer hardware that employs a specification associated with the object file to facilitate static checking of the object file, the checker providing information if a fault condition is determined, the specification specified at a source code level and stored in a specification repository (see, for example, column 16, lines 46-62, which shows employing a specification in the form of rules to facilitate static checking of the assemblies and providing information if a rule is violated, and column 10, lines 34-40, which shows that the rules are specified at a source code level, and see, for example, column 6, lines 47-53, which shows that the rules are stored in a repository).

With respect to claim 16 (original), the rejection of claim 15 is incorporated, and Cwalina further teaches the specification repository (see, for example, FIG. 1).

With respect to claim 17 (original), the claim is directed to a method of facilitating static checking of executable code, the elements of which are analogous to elements recited in claim 1 (see the rejection of claim 1 above).

With respect to claim 18 (original), the rejection of claim 17 is incorporated, and Cwalina further teaches removing the embedded specification from the executable code (see the rejection of claim 2 above).

With respect to claim 19 (original), the claim is directed to a computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 17 (see the rejection of claim 17 above).

With respect to claim 20 (currently amended), the claim is directed to a method of facilitating static checking of executable code, the elements of which are analogous to elements recited in claim 15 (see the rejection of claim 15 above).

With respect to claim 21 (original), the claim is directed to a computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 20 (see the rejection of claim 20 above).

With respect to claim 22 (currently amended), Cwalina teaches a computer-readable medium comprising computer executable instructions for transmitting a data packet transmitted between two or more computer components that facilitates static checking of executable code (see, for example, FIG. 2 and the abstract), the data packet comprising:

executable code having an embedded specification, the embedded specification providing information to be employed to statically check the executable code (see, for example, column 8, lines 20-21, which shows executable code in the form of a rule assembly and a target assembly, and column 8, lines 28-36, which shows that the assemblies have an embedded specification in

the form of rules, and see, for example, column 16, lines 46-62, which shows that the rules are employed to statically check the assemblies).

With respect to claim 23 (currently amended), Cwalina teaches a computer-readable medium comprising computer executable instructions for transmitting a data packet transmitted between two or more computer components that facilitates static checking of executable code (see, for example, FIG. 2 and the abstract), the data packet comprising:

a source-level specified specification that provides information to be employed to statically check the executable code (see, for example, column 16, lines 46-62, which shows a specification in the form of rules that are employed to statically check executable code in the form of assemblies, and column 10, lines 34-40, which shows that the rules are source-level specified).

With respect to claim 24 (original), the claim is directed to a computer readable medium storing computer executable components of an executable code check system, the elements of which are analogous to elements recited in claim 1 (see the rejection of claim 1 above).

With respect to claim 25 (original), the claim is directed to an executable code check system, the elements of which are analogous to elements recited in claim 1 (see the rejection of claim 1 above).

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

9. Claims 9 and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dollin, as applied to claim 1 above, in view of U.S. Pub. No. 2004/0230958 to Alaluf (art of record; "Alaluf").

With respect to claim 9 (currently amended), the rejection of claim 1 is incorporated. Dollin does not expressly disclose the object file being based, at least in part, upon a language that compiles to Common Language Runtime.

However, in an analogous art, Alaluf discloses performing static analysis on code that is based on a language that compiles to a Common Language Runtime (see, for example, paragraph [0038], lines 1-2, paragraph [0003], lines 1-3, and paragraph [0004], lines 8-9). Such code is platform and CPU independent (see, for example, paragraph [0003], lines 1-4, and paragraph [0004], lines 13-17).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the system of Dollin so as to operate on an object file that is based, at least in part, upon a language that compiles to a Common Language Runtime, as Alaluf suggests. For example, one of ordinary skill in the art would have been motivated to implement the system of Dollin such that it is platform and CPU independent.

With respect to claim 10 (previously presented), the rejection of claim 1 is incorporated. Dollin does not expressly disclose the object file being based, at least in part, upon at least one of C#, VISUAL BASIC.NET and Managed C++.

However, in an analogous art, Alaluf discloses performing static analysis on code that is based on C#, VISUAL BASIC.NET, or Managed C++, among others (see, for example, paragraph [0038], lines 1-2, and paragraph [0003], lines 8-11, and paragraph [0004], lines 8-9). Such code is platform and CPU independent (see, for example, paragraph [0003], lines 1-4, and paragraph [0004], lines 13-17).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the system of Dollin so as to operate on an object file that is based, at least in part, upon at least one of C#, VISUAL BASIC.NET and Managed C++, as Alaluf suggests. For example, one of ordinary skill in the art would have been motivated to implement the system of Dollin such that it is platform and CPU independent.

Conclusion

10. The prior art made of record and not relied upon is considered pertinent to Applicant's disclosure (see the attached Notice of References Cited).

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (571) 272-3707. The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Michael J. Yigdall
Examiner
Art Unit 2192

mjy

